Patent
Attorney Docket No. 030353

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	10/620,988
Applicant	:	Ron Everett
Filed	:	July 16, 2003
Title	:	DATA BASE AND KNOWLEDGE OPERATING SYSTEM
	:	
	:	
Confirmation No.	:	8827
Group Art Unit	:	2162
Examiner	:	Dennis Y. Myint
	:	
Docket No.	:	030353

RESPONSE TO OFFICE ACTION

September 15, 2006

Mail Stop - Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This is responsive to the Office Action dated June 15, 2006 in the above-referenced case. Please amend the application as indicated in the Listing of Claims. The Applicants request reconsideration of all currently pending claims in light of these Amendments and the Remarks following.

Listing of Claims

1. (Original) A data management system in a computing environment comprising:

    a. a data instance centric architecture;

    b. where each data instance is encapsulated in a common fundamental data structure; and

    c. where said common fundamental data structure also encapsulates references to associated separately encapsulated data instances.

2. (Original) The data management system of claim 1 wherein the said data-instance centric architecture and the said common fundamental data structure have structural symmetry.

3. (Original) The data management system of claim 1 wherein a first data instance is encapsulated with references to associated data instances and each of said associated data instances are separately encapsulated with a reference to said first encapsulated data instance.

4. (Original) The data management system of claim 3 wherein said data-instance centric architecture and the said fundamental data structure and the said encapsulated data instances and references have structural and relationship symmetry.

5. (Original) The data management system of claim 1 wherein a first data instance is encapsulated with references to associated data instances and each of said associated

data instances are separately encapsulated with a reference to said first encapsulated data instance; wherein each of said encapsulated references is a logical index which uniquely identifies each of said associated encapsulated data instances and also encodes the location of each of said associated encapsulated data instances; and wherein said logical index is 'm' dimensional, and has 'n' bits per dimension.

6. (Original) The data management system of claim 5 wherein said data instance centric architecture and said fundamental data structure; and the said encapsulated data instances and references have structural, relationship, value and containment symmetry.

7. (Original) The data management system of claim 1 wherein:

   a. said encapsulated references are in at least one dimensions; and

   b. each of said at least one dimensions corresponds to a type of association.

8. (Original) The data management system of claim 7 wherein each of said at least one dimensions has a plurality of said encapsulated references.

9. (Original) The data management system of claim 1 wherein the common fundamental data structure is application independent and is generally the same for all of said data instances.

10. (Original) The data management system of claim 2 wherein the common fundamental data structure is application independent and is generally the same for all of said data instances.

11. (Original) The data management system of claim 3 wherein the common fundamental data structure is application independent and is generally the same for all of said data instances.

12. (Original) The data management system of claim 4 wherein the common fundamental data structure is application independent and is generally the same for all of said data instances.

13. (Original) The data management system of claim 5 wherein the common fundamental data structure is application independent and is generally the same for all of said data instances.

14. (Original) The data management system of claim 6 wherein the common fundamental data structure is application independent and is generally the same for all of said data instances.

15. (Original) The data management system of claim 7 wherein the common fundamental data structure is application independent and is generally the same for all of said data instances.

16. (Original) The data management system of claim 8 wherein the common fundamental data structure is application independent and is generally the same for all of said data instances.

17. (Original) The data management system of claim 1 wherein at least one of said encapsulated references is a reference to an encapsulated data instance in another computing environment.

18. (Original) The data management system of claim 1 wherein said encapsulated references of at least one of said encapsulated data instances are unique and said encapsulated references of at least two of said encapsulated data instances are generally identical.

19. (Original) The data management system of claim 1 wherein said data instance centric architecture includes plurality of pre-existing encapsulated data instances, and said plurality of pre-existing encapsulated data instances have established associations, and at least one new encapsulated data instance is associated with at least one of said pre-existing encapsulated data instances.

20. (Original) The data management system of claim 1 wherein:

said data instance centric architecture includes a plurality of pre-existing encapsulated data instances; said encapsulated data instances have established associations; and wherein any of said pre-existing encapsulated data instances can be removed disassociated from other pre-existing associated encapsulated data instances.

21. (Original) The data management system of claim 1 wherein:

said data instance centric architecture includes a plurality pre-existing encapsulated data instances; said encapsulated data instances having established associations; wherein new associations between at least two pre-existing encapsulated data instances can be added.

22. (Original) The data management system of claim 1 wherein:

said data instance centric architecture includes a plurality of pre-existing encapsulated data instances; said encapsulated data instances having established associations; and wherein some of said pre-existing associations between said pre-existing encapsulated data instances can be removed.

23. (Original) The data management system of claim 1 further comprising:

a. finding specific unknown encapsulated data instances from a selection criteria of known encapsulated data instances by accessing said known encapsulated data instances representing said selection criteria;

b. accessing references encapsulated with said known encapsulated data instances representing said selection criteria;

c. using Boolean operations to compare said accessed encapsulated references to find references to said specific unknown encapsulated data instances; and

d. retrieving said specific unknown encapsulated data instances.

24. (Previously Presented) The data management system of claim 23 wherein:

a. said encapsulated references are embodied as logical indexes in a plurality of dimensions;

b. each of said dimensions corresponds to a type of association; and

c. said accessing further comprises accessing said encapsulated references from said dimensions specified in said selection criteria.

25. (Previously Presented) The data management system of claim 23 wherein:

said encapsulated references are 'm' dimensional logical indexes each of which uniquely identifies and encodes the location of said associated encapsulated data instances; and

further comprising filtering said encapsulated references by Boolean operations on at least one of said 'm' dimensional logical indexes.

26. (Previously Presented) The data management system of claim 24 wherein:

said encapsulated references are 'm' dimensional logical indexes each of which uniquely identifies and encodes the location of said associated encapsulated data instances; and

further comprising filtering said encapsulated references by Boolean operations on at least one of said 'm' dimensional logical indexes.

27. (Previously Presented) The data management system of claim 23 wherein said Boolean operations further comprise:

basic mathematical operators which result in the direct exclusion of at least one encapsulated reference from the result of said comparing in a single operation.

28. (Previously Presented) The data management system of claim 24 wherein said Boolean operations further comprise:

a basic mathematical operator which results in the direct exclusion of at least one encapsulated reference from the result of said comparing in a single operation.

29. (Previously Presented) The data management system of claim 25 wherein said Boolean operations further comprise:

a basic mathematical operator which results in the direct exclusion of at least one encapsulated reference from the result of said comparing in a single operation.

30. (Previously Presented) The data management system of claim 26 wherein said Boolean operations further comprise:

a basic mathematical operator which results in the direct exclusion of at least one encapsulated reference from the result of said comparing in a single operation.

31. (Original) The system of claim 1 wherein said encapsulated data instances have attributes of a user interface.

32. (Original) The system of claim 31 wherein said attributes of a user interface are selected from a group of user views, display elements, and data access methods.

33. (Original) The system of claim 1 further comprising searching said system wherein said encapsulated references of different said encapsulated data instances are used to derive desired results.

34. (Original) The system of claim 33 wherein said encapsulated references of different said encapsulated data instances are compared such for at least one of commonality, similarity and difference to derive sets of references corresponding to said desired results.

35. (Original) The system of claim 34 wherein said encapsulated references of different said encapsulated data instances are stored in an order based on value and are compared such for at least one of commonality, similarity and difference to derive sets of references corresponding to said desired results.

36. (Original) The system of claim 33 wherein:

a first data instance is encapsulated with references to associated data instances and each of said associated data instances are separately encapsulated with a reference to said first encapsulated data instance;

wherein each of said encapsulated references is a logical index which uniquely identifies each of said associated encapsulated data instances and also encodes the location of each of said associated encapsulated data instances; and

wherein said logical index is ‘m’ dimensional, and has ‘n’ bits per dimension;

said encapsulated references of different said encapsulated data instances are used by comparing such for at least one of commonality, similarity and difference to derive sets of references corresponding to said desired results.

37. (Original) The system of claim 33 wherein:

each of said at least one dimensions has a plurality of said encapsulated references; and

said encapsulated references of different of said encapsulated data instances are stored in an order based on value and are compared for at least one of commonality, similarity and difference to derive sets of references corresponding to said desired results.

38. (Cancelled)

39. (Cancelled)

40. (Original) The system of claim 1 further comprising:

encapsulated data instances representing ASCII characters;

said common fundamental data structures containing said encapsulated data instances representing ASCII characters also contain encapsulated references to encapsulated data instances containing said corresponding ASCII characters; and

said common fundamental data structures containing said encapsulated data instances containing said corresponding ASCII characters also contains encapsulated references to said encapsulated data instances representing corresponding ASCII characters.

41. (Original) The system of claim 40 wherein said encapsulated references with a given ASCII character data instance are references to other encapsulated data instances containing said ASCII characters based on position of said ASCII characters in the sequence of occurrence of said ASCII characters in said encapsulated data instances.

42. (Original) The system of claim 1 further comprising:

said encapsulated data instances representing Unicode characters;

said common fundamental data structures containing said encapsulated data instances representing Unicode characters also contain encapsulated references to encapsulated data instances containing said corresponding Unicode characters; and

said common fundamental data structures containing said encapsulated data instances representing Unicode characters also contains encapsulated references to said data instances representing corresponding Unicode characters.

43. (Original) The system of claim 42 wherein said encapsulated references with a given Unicode character data instance are references to other data instances containing said Unicode characters based on position of said Unicode characters in the sequence of occurrence of said Unicode characters in said encapsulated data instances.

44. (Original) The system of claim 1 wherein:

said encapsulated data instances comprise encapsulated data instances representing a token set of any data type;

said common fundamental data structures containing said data instances representing a token set of any data type also contain encapsulated references to encapsulated data instances containing said corresponding token set of any data type; and

said common fundamental data structures containing said encapsulated data instances representing token set of any data type also contains encapsulated references to said encapsulated data instances representing corresponding token set of any data type.

45. (Original) The system of claim 44 wherein said encapsulated references for a given token set of any data type data instance are references to other encapsulated data instances containing said token set of any data type based on position of token set of any data type in the sequence of occurrence of said token set of any data type in said encapsulated data instances.

46. (Original) The system claim 45 wherein:

said token set is selected from a group of a set of graphic descriptors, a set of colors, a set of shapes, a set of glyphs, a set of waveforms, a set of frequency values, a set of audio frequency values, a defined set of symbols, and real numbers.

47. (Original) The data management system of claim 1 wherein:

a. the common fundamental data structure is application independent and is generally the same for all of said data instances;
b. finding specific unknown encapsulated data instances from a selection criteria of known encapsulated data instances by accessing known encapsulated data instances representing said selection criteria;
c. accessing references encapsulated with said known encapsulated data instances representing said selection criteria;
d. using Boolean operations to compare said accessed encapsulated references to find references to said specific unknown encapsulated data instances; and
e. retrieving said specific unknown encapsulated data instances.

48. (Original) The system of claim 47 further comprising searching said system wherein said encapsulated references of different said encapsulated data instances are used to derive desired results.

49. (Original) The data management system of claim 1 wherein:

at least one of said encapsulated references is a reference to a encapsulated data instance in another computing environment; and

a first data instance is encapsulated with references to associated data instances and each of said associated data instances are separately encapsulated with a reference to an encapsulated data instance.

50. (Original) The data management system of claim 1 wherein:

a. said encapsulated references of at least one of said encapsulated data instances is unique and said encapsulated references of at least two of said encapsulated data instance are generally identical;

b. finding specific unknown encapsulated data instances from a selection criteria of known encapsulated data instances by accessing known encapsulated data instances representing said selection criteria;

c. accessing references encapsulated with said known encapsulated data instances representing said selection criteria;

d. using Boolean operations to compare said accessed encapsulated references to find references to said specific unknown encapsulated data instances; and

e. retrieving said specific unknown encapsulated data instances.

51. (Original) The data management system of claim 1 wherein:

said encapsulated references of at least one of said encapsulated data instances is unique and said encapsulated references of at least two of said encapsulated data instance are generally identical; and searching said system wherein said encapsulated references of different said encapsulated data instances are used to derive desired results.

52. (Original) A data management system in a computing environment comprising:

a. a data instance centric architecture;

b. where each data instance is encapsulated in a common fundamental data structure;

c. where said common fundamental data structure also encapsulates references to associated separately encapsulated data instances;

d. a first data instance is encapsulated with references to associated data instances and each of said associated data instances are separately encapsulated with a reference to said first encapsulated data instance;

e. each of said encapsulated references is a logical index which uniquely identifies each of said associated encapsulated data instances and also encodes the location of each of said associated encapsulated data instances;

f. said logical index is 'm' dimensional, and has 'n' bits per dimension; and

g. said encapsulated references are in at least one dimensions; and

h. each of said at least one dimensions corresponds to a type of association.

53. (Original) The data management system of claim 52 wherein the common fundamental data structure is application independent and is generally the same for all of said data instances.

54. (Original) The data management system of claim 53 further comprising:

a. finding specific unknown encapsulated data instances from a selection criteria of known encapsulated data instances by accessing known encapsulated data instances representing said selection criteria;

b. accessing references encapsulated with said known encapsulated data instances representing said selection criteria;

c. using Boolean operations to compare said accessed encapsulated references to find references to said specific unknown encapsulated data instances; and

d. retrieving said specific unknown encapsulated data instances.

55. (Original) The system of claim 54 further comprising searching said system wherein said encapsulated references of different said encapsulated data instances are used to derive desired results.

56. (Original) The data management system of claim 55 wherein at least one of said encapsulated references is a reference to a encapsulated data instance in another computing environment.

57. (Original) The data management system of claim 56 wherein said encapsulated references of at least one of said encapsulated data instances is unique and said encapsulated references of at least two of said encapsulated data instances are generally identical.

58. (Original) The system of claim 57 further comprising searching said system wherein said encapsulated references of different said encapsulated data instances are used to derive desired results.

59. (Original) The data management system of claim 52 further comprising:

a. finding specific unknown encapsulated data instances from a selection criteria of known encapsulated data instances by accessing known encapsulated data instances representing said selection criteria;

b. accessing references encapsulated with said known encapsulated data instances representing said selection criteria;

c. using Boolean operations to compare said accessed encapsulated references to find references to said specific unknown encapsulated data instances; and

d. retrieving said specific unknown encapsulated data instances.

60. (Original) The system of claim 52 further comprising searching said system wherein said encapsulated references of different said encapsulated data instances are used to derive desired results.

61. (Original) The data management system of claim 52 wherein at least one of said encapsulated references is a reference to a encapsulated data instance in another computing environment.

62. (Original) The data management system of claim 52 wherein said encapsulated references of at least one of said encapsulated data instances is unique and said encapsulated references of at least two of said encapsulated data instance are generally identical.

63. (Withdrawn) A method to coordinating physical memory addressing and logical memory addressing in an encapsulated data instance centric architecture comprising:

   a. corresponding to each encapsulated data instance there is a logical reference to said encapsulated data instance;
   b. encapsulating said logical reference to said encapsulated data instance in a first container;
   c. relating said logical reference in said first container with a physical reference to a location where said encapsulated data instance is stored in a physical storage medium;
   d. encapsulating said physical reference in a second container; and
   e. relating said physical reference in said second container with said logical reference to said encapsulated data instance in said first container.

64. (Withdrawn) The method of claim 63 wherein said container is a fundamental data structure.

65. (Withdrawn) The method of claim 63 wherein said physical reference is 'n' dimensional, the number of dimensions and the number of bits in each dimension correspond to the structure of said physical storage medium.

66. (Withdrawn) The method of claim 65 wherein using said physical reference to calculate an address in said physical storage medium.

67. (Withdrawn) The method of claim 63 further comprising:

   a. coordinating a plurality of data instances; and
   b. encapsulating a plurality of said logical references and a plurality of said physical references in respective said first and second containers.

68. (Withdrawn) The method of claim 63 further comprising sorting said plurality of said logical references in said first containers.

69. (Withdrawn) The method of claim 68 further comprising sorting said plurality of said physical references in said second containers.

70. (Withdrawn) A method for managing data storage in a data instance centric architecture having a plurality of variable length data instances comprising:

   a. storing said data instances generally sequentially in a physical storage medium;
   b. storing each data instance in a respective allocated space;
   c. updating one of said data instances;

d. integrating said updated data instance into said physical storage medium by determining the amount of physical space that is needed to store said updated data instance;

e. when said physical space is equal to or less than said respective allocated space then storing said updated data instance in the said respective allocated space;

f. when said physical space is greater than said respective allocated space then identifying at least one physically adjacent data instance having an aggregate allocation space equal to or greater than the difference between said physical space and said respective allocated space; and

g. writing the said updated data instance to said physical storage medium at an updated location based on the size and number of said adjacent data instances.

71. (Withdrawn) The method of claim 70 for managing data storage further comprising:

writing the said updated data instance to said physical storage medium in an updated respective allocated space based on the size and number of said adjacent data instances.

72. (Withdrawn) The method of claim 70 for managing data storage further comprising:

moving said at least one physically adjacent data instance to a location after a last stored data instance; and writing the said updated data instance to said physical storage medium to a location of said at least one physically adjacent data instance and said respective data instance.

73. (Withdrawn) The method of claim 70 for managing data storage further comprising:

moving said at least one physically adjacent data instance to a location after a last stored data instance; and

writing the said updated data instance to said physical storage medium in an updated allocation space equal to the aggregate of said respective allocated space and the allocated space of the said moved at least one physically adjacent data instances.

74. (Withdrawn) The method of claim 73 for managing data storage wherein the said updated allocation space is greater than the said physical space of the said updated data instance.

75. (Withdrawn) The method of claim 70 for managing data storage wherein said at least one physically adjacent data instance occur sequentially after said updated data instance.

76. (Withdrawn) The method of claim 70 for managing data storage wherein said at least one physically adjacent data instance occur sequentially before said updated data instance.

77. (Withdrawn) The method of claim 70 for managing data storage wherein said at least one physically adjacent data instance occur sequentially both before and after said updated data instance.

78. (Withdrawn) The method of claim 70 for managing data storage wherein:

whichever of said at least one physically adjacent data instance has an allocated space that is closest to and greater than the said difference between said physical space and said respective allocated space of the said updated data instance the said at least one physically adjacent data instance is moved to a location after a last stored data instance;

writing the said updated data instance to said physical storage medium to a location of said at least one physically adjacent data instance and said respective data instance; and

writing the said updated data instance to said physical storage medium in an updated allocation space equal to the aggregate of said respective allocated space and the allocated space of the said moved at least one physically adjacent data instances.

79. (Withdrawn) The method of claim 70 for managing data storage wherein when said physical space of said updated data instance is less than the allocated space of said at least one physically adjacent data instance then the updated data instance is moved to a location after a last stored data instance.

80. (Withdrawn) The method of claim 79 for managing data storage wherein the said respective allocated space of the said moved updated data instance is added to the allocated space of the physically adjacent data instance sequentially before said updated data instance.

81. (Withdrawn) The method of claim 70 for managing data storage wherein at least one of said allocated space is greater than the size of the respective data instance.

82. (Currently Amended) A method to convert a non-data instance centric database to a data instance centric database comprising:

a. creating encapsulated data instances in said data instance centric database representing elements of said non-data-instance centric database schema and data elements of said non-data-instance centric database; ~~and~~

b. ~~create~~ creating associations amongst the said data instances in said data instance centric database representing the relationships between said data elements and said schema elements of the non-data-instance centric database and storing said associations within each associated encapsulated data instance.

83. (Original) The method of claim 82 wherein said converting is through a software agent which is a data instance in said data instance centric database.

84. (Original) The method of claim 82 wherein said non-data instance centric database includes a flat file.

85. (Original) A data management system comprising:

a. one or more items;

b. wherein each of said items encapsulates a data instance; and

c. wherein items which are associated with each other encapsulate mutual references to each other.

86. (Original) The data management system of claim 85 wherein each of said items is represented in a fundamental data structure.

87. (Original) The data management system of claim 85 wherein each of said items has a unique reference associated therewith.

88. (Original) The data management system of claim 87 wherein said unique reference also serves as an index to physically locate said data instance associated with each of said items.

89. (Original) The data management system of claim 85 wherein said references to associated items are arranged in sets defining the type of association between said item and each of said other items referenced in said set.

90. (Original) The data management system of claim 87 wherein each of said references is an "m" dimensional index, each of said dimensions being "n" bits in length.

91. (Original) The data management system of claim 90 wherein "m" is 4 and "n" is 30.

92. (Original) The data management system of claim 85 wherein said items may act as containers for one or more other member items.

93. (Original) The data management system of claim 92 wherein membership of an item within a container item is indicated by an identity in one or more of said "m" dimensions in said logical index of said container item and each of said member items.

94. (Original) The data management system of claim 85 wherein each of said items may encapsulate embedded elements.

95. (Original) The data management system of claim 94 wherein said embedded elements are references to other items.

96. (Original) The data management system of claim 85 wherein said data instances may contain data of any type.

Remarks

Claims 1-13, 40-62, and 81-96 are pending in the application. Claims 1-24, 27, 28, 31-34, 36, 40, 42, 44, 47-62, 82-90, and 92-96 have been rejected. Claims 25, 26, 29, 30, 35, 37, 41, 43, 45, 46 and 91 have been objected to, but the Examiner has indicated that these claims would be allowed if rewritten in independent form. The Applicant declines to do so at this time, pending the outcome of the prosecution of this case.

The Examiner has addressed the Applicant's previous arguments in the Response to Arguments section of the current Office Action. Therefore, the Applicant will first address the Examiner's response. With respect to Claim 1, which had been rejected by the Examiner in view of U.S. Patent No. 6,609,132 ("White, et al., hereinafter "White"), the Examiner states that the argument presented in the previous Response is directed to only one specific embodiment of several preferable embodiments which White presents. The Examiner further states that, to the contrary of the Applicant's arguments, White teaches features for encapsulating relationships between data instances and specifically quotes column 6, lines 66, through column 7, line 11 of the White reference, which discloses that data structures holding the objects of the database have a plurality of attributes as data members for storing useful information that describes characteristics of the corresponding object. In the Examiner's interpretation of White, the attributes of a given object may be used to store information regarding relationships between associated data objects.

The Applicant strongly disagrees with the Examiner's interpretation of White. The Applicant interprets White as a system wherein the relationships between data objects are required to be kept separate from the actual data objects. The Applicant draws the Examiner's attention to column 5, line 1 through column 6, lines 62 which is entitled "Generalized

Embodiment of the Software Application of the Present Invention". The information given in this passage describes a generalized embodiment of the invention, the characteristics of which also apply to any specific embodiments which are presented in the patent. In at least two separate portions of the description of the generalized embodiment of the invention, it describes wherein the relationship information is kept separate from the data information. The Applicant draws the Examiner's attention to column 5, lines 19-26 which states as follows:

> Each relation is represented by a data structure that stores textual annotation characterizing the semantics of a relationship linking two (or more) objects. *Preferably, the data structure (i.e., data records) representing a given relation linking two or more objects are separate from (and indirectly coupled to) the data structures representing these two or more objects.* (emphasis added)

This concept is reinforced later in the passage, starting at column 6, line 9 and extending to Column 6, line 22.

> Moreover, *the data structures (i.e., data records) representing the given relation is preferably separate from (and indirectly coupled to) the data structures representing the subject object(s) and direct object(s);* thus, in this case, the bi-directional modifier text is not defined by (and thus can differ from) any fields (attributes) of the data structures that represent the subject object(s) and direct object(s) of the given relation. This indirect coupling enables a relation to characterize the semantics of multiple relationships linking two (or more) objects (thus saving storage spaces) and enables a relation to characterize the semantics of a relationships linking two (or more) objects in disparate systems (for example, two different databases). (emphasis added)

The only specific embodiment of the invention disclosed in the patent appears to be shown in Figures 2-3 and is described beginning in column 6, line 65 in a section entitled "Illustrative Embodiments of the Logical Data Structures of the Object Data Model of the Present Invention". This description extends to column 10, where a mechanism for viewing and navigating the object data model is presented. The only other matter disclosed within the patent begins in column 20, where an illustrative embodiment of a graphical user interface for creating and updating the model is presented. As a result, the only specific embodiment of the generalized invention is described beginning in column 6, line 65 and as shown in Figures 2-4.

The specific embodiment described and shown in Figures 2-4 clearly show that the relationship information regarding two or more objects is kept separately from those objects and not encapsulated therewith, thereby conforming to the generalized embodiment previously described. The Examiner seems to indicate that the attribute portion of each data object may be used to store relationship information. This goes against the teaching of White and more specifically, White explicitly discloses only two purposes for the attribute entries in each data object. The Applicant directs the Examiner's attention to column 7, line 5 through column 7, line 8, which states as follows:

> The attributes of a given object may be used *to encapsulate data and/or link to software functionality and/or processes* pertinent to the given object. (emphasis added)

White does not state that the attributes of the data objects may be used for the purpose of storing relationship information, and as discussed with respect to the generalized embodiment, states exactly the opposite. White is completely devoid of any disclosure of the use of the attributes for the storage of relationship information, and any attempt by the Examiner to read this feature into the invention is hindsight construction by the Examiner of White in a manner not contemplated by the inventor. As such the Applicant respectfully submits that the White reference cannot and should not be interpreted in this manner. The Examiner provides an indication of the creative interpretation of White when he states on page 3 of the Office Action "The above reference clearly indicates data objects in White's method/system could encapsulate more data objects inside and/or pointers to separately encapsulated data objects (data instances), which anticipates the limitation (c) of claim 1 of the application." However, as pointed out by the Applicant, White does not provide a disclosure of the use of the attributes of a data object for this purpose and the Examiner should not be permitted to read that particular use of the data object attributes into White.

Furthermore, even if the attribute portion of each data object were used to encapsulate data objects and/or pointers to data objects therein, there is no mention in the Examiner's interpretation of White wherein the *type* of relation between the present data object and the referenced data object is stored. Therefore, such a construction does not work under the model presented by White as interpreted by the Examiner.

The Examiner further cites a portion of White on the top of page 4 of the Office Action which appears to discuss entries in the relation table, which is shown in Figure 3 of White and the relation object table entry which is shown in Figure 2 of White, both of which clearly indicate that the relationship information is kept separately from the objects. The Applicant directs the Examiner's attention to Figures 2 and 3; both figures show objects (a, b, c, d) down the right hand side of the respective figures. The Applicant points out that arrows do not extend from any of objects a, b, c, d to any other of the objects a, b, c, d. Therefore, this indicates that the attribute portion of the data objects is not used for the purpose of storing relationship information. The portion of White cited by the Examiner at the beginning of page 4 of the Office Action clearly indicates that the relation type and the actual relations linking the data objects are kept separately in tables and not as encapsulated attributes of objects, such as objects a, b, c, d shown in Figures 2 and 3.

The Examiner further argues on page 4 of the Office Action that the specification of the present application recites data structures for the storage of relationship information which are conceptually similar to tables which are used in White, and references specifically paragraph 91 of the present application. However, paragraph 91 describes an indexing system wherein the data objects of the present application are identified by a unique multidimensional index referred to in the specification as {E,R,C,I}. The cited paragraph has nothing to do with any relationship between data structures and tables. The Examiner states that the application recites data structures, which are conceptually similar to tables, which White uses to store the relation type information and relation object information. Even if the Examiner's statement could somehow be construed as relevant (*i.e.*, the data structures are conceptually similar to tables) the fact remains that White discloses the separate storage of the object information and the relationship information, regardless of whether stored in data structures or tables. As a result, element (c.) of claim 1, which states that the relation information is encapsulated within each data structure storing the data objects is not met by White.

The Examiner has rejected Claims 1-4, 7, 9-16, 53, 85-87, 89 and 92 under 35 U.S.C. § 102(e) as being anticipated by White.

With respect to Claim 1, the Applicant respectfully submits that element (a.), "a data instance centric architecture" is not disclosed by White. As defined in the present application, a data instance centric architecture requires that the encapsulated data instances make up the entirety of the database. No auxiliary information or tables are needed in which data objects, relationship predicates, relation types or data types are stored. White discloses a system which includes data objects wherein the information regarding the relationships between data objects is stored separately. Therefore, White does not represent a data instant centric architecture as that term is defined in the present application.

The Examiner further states that element (b.) of Claim 1 is disclosed in White. Element (b.) reads "where each data instance is encapsulated in the common fundamental data structure". The Applicant respectfully submits that to be encapsulated in a common fundamental data structure, as defined in the present application, the data instance must include all information regarding that data instance because there is no other place in the model of the present application in which to store the relationship information. As a result, the Applicant respectfully submits that the model disclosed in White does not have encapsulated data instances, but instead, uses auxiliary tables or data structures to include information regarding the relationships between data instances. As a result, White also lacks the disclosure of the use of a common fundamental data structure in which to encapsulate data instances, because the relation information must be stored in either a table or a database structure. Those database structures would not be implemented using the common fundamental data structures of the same type utilized to encapsulate the data instances because different types of information is being encapsulated therein. As a result, White also lacks a disclosure of the use of common fundamental data structures in which to encapsulate the data instances.

With respect to element (c.) of Claim 1, the Applicant's discussion above regarding the storage of relationship information apart from data instances shows that element (c.) of Claim 1 is not met by White. Element (c.) states " wherein said common fundamental data structure also encapsulates references to associated separately encapsulated data instances". As discussed above, this limitation is not met by White because the relation information is kept separately from the data structures containing the data instances and no embodiment of White shows a contrary model. As a result, the Applicant respectfully submits that none of the elements of Claim 1 are shown by White and, as a result, requests that the Examiner withdraw the rejection of Claim 1 under § 102(e).

All other currently pending claims of the application, with the exception of Claims 82-84, either depend from Claim 1 or contain limitations very similar to Claim 1 and have been rejected on the same basis as with respect to Claim 1. In addition, the Applicant has provided in a previous Office Action remarks distinguishing these other claims from White and those will not be repeated here for the sake of brevity. An allowance by the Examiner of Claim 1 will result in the allowance of all claims dependent from Claim 1 and other independent claims having similar limitations of Claim 1.

Claims 82-84 have been rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Published Patent Application No. 2005/0044079 (Abineri, et al., hereinafter "Abineri"). Claims 82-84 claim a method of converting a non-data instant centric database to the data instance centric base of the present application. Abineri discloses in paragraph [0106] the conversion of a flat file type database to an object oriented style database. However, it does not provide any details of how the conversion is to be executed. Abineri only expresses the need to do the conversion to meet the needs of the main objective of Abineri, which is to provide a visual representation of the data in the database. Abineri, however, does not provide a method of creating data instances in a data instant centric architecture, because the data instant centric architecture is not defined in Abineri. In the present application, the data instant centric database schema is clearly defined as having associations stored within each of the associated encapsulated data instances. As a result, the Applicant has amended Claim 82 to make it clear that after associations are created, they are stored in the encapsulated data instances which are thereby associated with each other. Neither Abineri nor White disclose this step because neither of the cited references discloses a schema wherein data instances are encapsulated with all relevant information regarding that data object. *i.e* , no other types of objects exist in the database other than those based on common fundamental data structure representing a data instance and no additional tables are necessary to encode auxiliary information such as relationships and associations).

The Examiner has rejected Claims 5, 6, 8, 18, 19-24, 31-34, 36, 47-48, 50-52, 54-55, 58-60, 62, 88, 90 and 93 under 35 U.S.C. § 103(a) as being unpatentable over White in view of U.S. Patent No. 5,809,297 (Kroenke, et al., hereinafter "Kroenke"). With respect to the combination of White and Kroenke, the Applicant respectfully submits that there is no motivation to combine White with Kroenke as discussed in the previous response. White teaches away from the present application as previously discussed with respect to Claim 1 in that it teaches storing the relationship information separate from the data objects. Additionally, Kroenke also teaches away from the present application because Kroenke teaches a computer based system for allowing a user to create a relational database schema. Because the present application discloses the data instant centric architecture, a system allowing the user to create relational database schema would teach away therefrom because the object of a data instant centric architecture is to avoid the use of relational type database tables. As a result, there is no motivation to combine White and Kroenke because both appear to teach solutions directly in opposition to those sought in the present application. Further, the combination of White and Kroenke does not disclose the presently claimed invention as White does not teach any of the elements of Claim 1. Therefore, the combination of White and Kroenke does not disclose all elements of the claims of the present application. Specific elements which the user says are disclosed in Kroenke have been discussed in the previous response and will not be recited again for the sake of brevity. However, in any case, the inclusion of White in the combination precludes the disclosure of the invention by the combination of White and Kroenke regardless of what is disclosed in Kroenke.

The Examiner has rejected Claims 27 and 28 under 35 U.S.C. § 103(a) as being unpatentable over White in view of Kroenke and further in view of U.S. Published Patent Application No. 2003/0216169 (Walker, et al., hereinafter "Walker"). Walker is cited for its disclosure of Boolean operations as basic mathematical operators. The same comments with respect to White and Kroenke apply hear as well. The Applicant respectfully submits that their combination is not only improper but does not disclose all elements of the claimed invention.

Claim 40 has been rejected under 35 U.S.C. § 103(a) as being unpatentable over White in view of U.S. Patent No. 5,873,049 (Bielak, et al., hereinafter "Bielak"). The Examiner states that White does not explicitly disclose the limitation of "encapsulated data instances representing ASCII characters" but that Bielak teaches those limitations. The Applicant respectfully submits that White not only does not teach encapsulated data instances representing ASCII characters, it does not teach encapsulated data instances at all regardless of whether the data therein represents ASCII characters or any other type of information. In addition, Claim 40 states that the common fundamental data structures containing the data representing ASCII characters also contain encapsulated references to data instances containing or referencing those ASCII characters and vice versa. As pointed out in the previous Response, Bielak teaches a system for the storage of seismic data in an ASCII format. It does not explicitly teach where individual ASCII characters are stored separately in encapsulated data objects and, in particular, does not teach wherein all other data objects utilizing that ASCII character have a reference thereto and vice versa. Furthermore, as previously discussed, White teaches away from the present application in that White teaches the separate storage of relationship information and data object information and, therefore, teaches away from encapsulated data objects. As a result, the Applicant respectfully submits that not only is the combination of White and Bielak improper, but that the recited combination does not disclose all elements of Claim 40.

The Examiner has rejected Claim 42 under 35 U.S.C. § 103(a) as being unpatentable over White in view of U.S. Published Patent Application 2003/0076978 (Eversole, et al., hereinafter "Eversole"). Claim 42 states that the common fundamental data structures containing the encapsulated data instances represent uni-code characters This is basically the same rejection as with respect to Claim 40 involving uni-code characters instead of ASCII characters. The Applicant's interpretation of Eversole is that Eversole merely suggests that an object property type can be identified as a uni-code string, not that uni-code characters can be encapsulated in common fundamental data structures of the present application. Therefore, in addition to the points discussed above wherein White teaches away from the present application and therefore renders the combination improper, the combination of White and Eversole does not suggest or teach all elements of Claim 42.

The Examiner has rejected Claim 44 under 35 U.S.C. § 103(a) as being unpatentable over White in view of U.S. Patent No. 5,812,840 ("Shwartz"). The Examiner states that White does not disclose the encapsulated data instances which represent a token set of any data type but that Shwartz teaches a method and system for a database including this limitation. The Applicant respectfully submits that Shwartz appears to teach a system which is able to assist the user in constructing syntactically and semantically valid SQL statements. Claim 44 adds to the limitations of Claim 1 that encapsulated references to other data instances may be grouped into token sets containing references of any given type. This is the mechanism whereby the present application defines the type of association between an encapsulated data object and those data objects referenced by the encapsulated data object. The groupings of data objects indicate the same type of relationship between the a particular data object and the grouped references to other data objects. That is, all data instances which are members of the token set encapsulated in any particular data instance indicate that they all have the same relationship to the encapsulating data instance. This is not disclosed in Shwartz and, as discussed above, White teaches away from the present invention, rendering the combination of White and Shwartz and its application to the present application improper. Further, the combination of White and Shwartz does not teach all elements of the present application.

Claims 17, 49 and 61 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over White in view of U.S. Patent No. 6,957,214 ("Silberberg, et al., hereinafter "Silberberg"). The Examiner states that White does not explicitly teach having a reference to an encapsulated data instance in another computing environment. As stated in the previous Response, the Applicant submits that Silberberg teaches accessing data in a plurality of domains which are distributed, but fails to teach that the references are encapsulated within other encapsulated data instances that are associated therewith. As with the other rejections under § 103(a), the Applicant submits that White teaches away from the present application rendering its combination with Silberberg improper and further that the combination of White and Silberberg do not teach all elements of Claims 17, 49 and 61.

Claims 94-96 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over White in view of U.S. Patent No. 6,016,497 ("Suver"). The Examiner states that White does not explicitly teach an encapsulation of embedded items but that Suver teaches a system and method for storing and accessing embedding information in an object relational database. Suver teaches the embedding of subtables within a relational database system, not the embedding of items in an encapsulated data instance, which is encapsulated in the common fundamental data structure. Furthermore, as previously stated, White teaches away from the present application rendering its combination with Suver improper and further the combination of White and Suver does not explicitly teach all elements of Claims 94-96.

The Applicant appreciates the Examiner's acknowledgement that Claims 25-26, 29-30, 35, 37, 41, 43, 45-46 and 91 would be allowable if rewritten in independent form. However, the Applicant declines to do so pending the outcome of the prosecution of the remaining claims.

Conclusion

The Applicant has again put forth reasoned arguments in support of the patentability of the rejected claims and has shown how White teaches away from the present application in that White requires that the relationship information between data objects be stored separately from the data objects while the present application requires that that information be encapsulated within the common fundamental data structure containing any particular data instance. As such, all rejections based on White should be traversed and those claims should be allowable.

The Applicant believes that no additional fee is required with this Response. However, if any other fees are required for any reason, the Commissioner is hereby authorized to charge Deposit Account No. 02-4800 the necessary amount.

Should any issues remain, the Examiner is invited to contact the undersigned at the number listed below to advance prosecution of the case.

Respectfully submitted,

Dennis M. Carleton
Registration No. 40,938

BUCHANAN INGERSOLL & ROONEY PC
20th Floor, One Oxford Centre
301 Grant Street
Pittsburgh, Pennsylvania 15219-1410
Phone: 412-562-1895
Fax: 412-562-1041
e-mail: dennis.carleton@bipc.com
Attorneys for Applicant(s)